Extension of the Genetic Algorithm Based Malware Strategy Evolution Forecasting Model for Botnet Strategy Evolution Modeling

Nikolaj Goranin
Information Security Laboratory, Department of Information System
Faculty of Fundamental Sciences, Vilnius Gediminas Technical University
Saulėtekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania

ngrnn@fmf.vgtu.lt

Antanas Cenys
Information Security Laboratory, Department of Information System
Faculty of Fundamental Sciences, Vilnius Gediminas Technical University
Saulėtekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania

ac@fmf.vgtu.lt

Jonas Juknius
Information Security Laboratory, Department of Information System
Faculty of Fundamental Sciences, Vilnius Gediminas Technical University
Saulėtekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania
The Communications Regulatory Authority of the Republic of Lithuania
Algirdo Str. 27A, LT-03219 Vilnius, Lithuania

jjj@cert.lt

ABSTRACT

Botnets are considered to be among the biggest current threats to global IT infrastructure. Botnets are rapidly evolving and forecasting their survivability and propagation strategies is important for development of countermeasure techniques. Existing malware propagation models mainly concentrate on malware epidemic consequences modeling, i.e. forecasting the number of infected computers, simulating malware behavior or economic propagation aspects and are based only on current malware propagation strategies or oriented to other malware types. In this article we propose the botnet-oriented extension to our genetic algorithm based model, which aims at forecasting botnet propagation strategy evolution and may be used as a framework for other characteristics evolution forecasting. The efficiency of strategies is evaluated by applying the proposed fitness function. Genetic algorithm is selected as a modeling tool taking into consideration the efficiency of this method while solving optimization and modeling problems with large solution space. The main application of the proposed model framework is a countermeasures planning in advance and computer network design optimization.

1.0 INTRODUCTION

The term bot describes a remote control program loaded on a computer, usually after a successful invasion (compromise can be achieved with the help of a worms, Trojan horses or “backdoor” software [1]), that is often used for nefarious purposes [2] usually against the computer owners’ intentions and without their knowledge [3]. A botnet is a network of computers on which a bot has been installed, and is usually managed remotely from a Command & Control (C&C) server. The main purpose of botnets is to use hijacked computers for fraudulent online activities [4]: identity theft [5], sending spam, performing

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2010	**N/A**	**-**

4. TITLE AND SUBTITLE
Extension of the Genetic Algorithm Based Malware Strategy Evolution Forecasting Model for Botnet Strategy Evolution Modeling

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Information Security Laboratory, Department of Information System Faculty of Fundamental Sciences, Vilnius Gediminas Technical University Saulëtekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)
11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA564697. Information Assurance and Cyber Defence (Assurance de l'information et cyberdefense). RTO-MP-IST-091

14. ABSTRACT
Botnets are considered to be among the biggest current threats to global IT infrastructure. Botnets are rapidly evolving and forecasting their survivability and propagation strategies is important for development of countermeasure techniques. Existing malware propagation models mainly concentrate on malware epidemic consequences modeling, i.e. forecasting the number of infected computers, simulating malware behavior or economic propagation aspects and are based only on current malware propagation strategies or oriented to other malware types. In this article we propose the botnet-oriented extension to our genetic algorithm based model, which aims at forecasting botnet propagation strategy evolution and may be used as a framework for other characteristics evolution forecasting. The efficiency of strategies is evaluated by applying the proposed fitness function. Genetic algorithm is selected as a modeling tool taking into consideration the efficiency of this method while solving optimization and modeling problems with large solution space. The main application of the proposed model framework is a countermeasures planning in advance and computer network design optimization.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**20**	

massive, sophisticated, tactically agile, difficult to trace denial-of-service attacks [3],[6] which due to botnet development have gone evolution from theoretical to real informational weapons [7], click fraud, key cracking, phishing, distribution of new malware to the wild [3], distribute pirated media and other tasks [8]. Botnets are managed by a criminal, a group of criminals or an organized crime syndicate [4] which are called botmasters or botherders [6].

Botnets dominate today's attack landscape [9] and it is widely accepted that botnets pose one of the most significant and steadily increasing threats to the Internet with devastating consequences [4],[10]. Bot technology has accelerated in its development in the last few years [6]. The reason for this change is a significant shift in motivation for malicious activity has taken place over the past several years: from vandalism and recognition in the hacker community, to attacks and intrusions for financial gain. This shift has been marked by a growing sophistication in the tools and methods used to conduct attacks, thereby escalating the network security arms race [5]. Bot armies are effective for two reasons: they can execute multiple overt actions against targets and can, alternatively, provide multiple coordinated and covert listening points within targeted networks and computer systems [6]. It is a main weapon used on targeted computers and also a significant threat even on a whole country scale, when botnets are used in cyber warfare as brute force army performing Distributed Denial of Service attacks [11].

It is difficult to measure the extent of damage caused on the Internet by botnets, but it is widely accepted that the damage done is significant [12]. Rajab et al. find [10] that a major contributor of unwanted Internet traffic - 27% of all malicious connection attempts - can be directly attributed to botnet-related spreading activity [13]. The botnet size estimation vary a lot, especially due to the fact that hackers frequently attack large numbers of easy-to-compromise home computers [14], the number of which can not be measured accurately (they are not constantly on-line, no registration in DNS servers, dynamic IP's, etc.) [15]. For example between July 1 and December 31, 2007, Symantec observed an average of 61,940 active bot-infected computers per day, a 17 percent increase from the previous reporting period [16]; Zhuge et al. [15] state that they tracked 3,290 IRC-based botnets during the measurement period between June 2006 and June 2007 and in total observed about 700,700 distinct IP addresses, the biggest botnet observed they tracked has controlled more than 50,000 hosts. According to Enisa report [4] each botnet has an average of 20.000 compromised computers (bots): some C&C servers manage just a few infected computers (~10), large ones manage thousands of bots (~300.000). Our measurements performed at The Communications Regulatory Authority of the Republic of Lithuania has show the following dynamics: for year 2007 the total number of unique bots was equal to 3917, for 2008 – 1502 (due to effective measures taken against centralized botnets), and even 10690 for just three first months of 2009.

The deployment and operation of bot armies are aided by the security vulnerabilities that exist in contemporary software; vulnerabilities that are likely to increase in number commensurately with the increase in the size of software products [6]. The trend toward an economic motivation is likely to catalyze development of new capabilities in botnet code making the task of securing networks against this threat much more difficult [5]. According to ENISA report [4] the objective of criminals using botnets will be twofold: to increase the number of infected computers and increase the stealth (or as we propose using the term survivability) by different methods – polymorphism, P2P, active protection mechanisms. It is important to outline and predict the future botnet techniques since this allows researches to develop techniques against these kinds of botnets before they appear in the wild [17]. That is why in this article we propose using an extension for our previously described [18] genetic algorithm based malware strategy forecasting model for predicting techniques the future botnets will use to improve their survivability, propagation and other characteristics. Existing malware propagation models mainly concentrate on malware epidemic consequences modeling, i.e. forecasting the number of infected computers, and are based only on current malware propagation strategies or oriented to other malware types. Genetic algorithm [19] was selected as a modeling tool since it simulates natural selection by means of repeatedly evolving population of solutions (malware propagation and survivability strategies in our case) and therefore may be used for predicting and modeling possible future propagation and survivability strategies.

The main application of the proposed model is a countermeasures planning in advance and computer network design optimization.

The article is organized as follows: Chapter 1 – Introduction, provides description of problem area; Chapter 2 – Botnet Strategy Analysis and Examples, provides technical analysis of modern botnets, which is required to explain the changes made to the malware propagation model first described in [18]. Analysis is also very important since effective network security is based on the detailed understanding of the mechanisms used by malware [5]; Chapter 3 – Prior and Related Work, describes different malware propagation models; Chapter 4 – Botnet Strategy Evolution Forecasting Model is describing the extension to the GA based malware propagation model, its limitations and results; Chapter 5 – Conclusions.

2.0 BOTNET STRATEGY ANALYSIS AND EXAMPLES

2.1 Botnet Strategy Analysis

As we have defined in [18] strategy is a combination of methods and techniques, used by malware to achieve the tasks assigned to it by the creator. Over the years botnet capability has increased substantially to the point of blurring the lines between traditional categories of malware [5] and in fact possesses characteristics are those of a virus, a worm, and a Trojan [6]. While bots belonging to a certain botnet are expected to have some distinct modes of operation, individual bots are also expected to have unique behaviors due to variabilities in the software or hardware they run on, phase difference in their states, different background applications running simultaneously etc. [8]. Botnets are different from traditional discrete infections in that they act as a coordinated attacking group [20]. We would even like to propose understanding of a botnet as a single artificial organism, not a combination of small organisms (bots), since the task being assigned is assigned to the whole botnet. Botnet has similarities in organization with termite or bee families, where a single individual is not very important (except dominant), but the whole family forms a real power, which some scientists are ready to name as a super-organism.

The overall architecture and implementation of botnets is complex, and is evolving toward the use of common software engineering techniques such as modularity [5]. The taxonomy proposed in [5] classifies botnets by key mechanisms used: (1) architecture, (2) botnet control mechanisms, (3) host control mechanisms, (4) propagation mechanisms, (5) target exploits and attack mechanisms, (6) malware delivery mechanisms, (7) obfuscation methods, and (8) deception strategies. On the other hand according to [6] the botnet creation process can be described in 7 steps: (1) malware creation, (2) command and control creation, (3) malware propagation, (4) malware infestation, (5) command and control setup, (6) further malware download, and (7) malware check-in for further instructions via the command and control setup.

Most botnets that have appeared prior to 2005 have had a common centralized architecture. That is, bots in the botnet connect directly to some special hosts (called "command-and-control" servers, or "C&C" servers) [21] and were based on IRC due to its ability to scale to thousands of clients easily [8]. According to ENISA report [4] IRC is still being used by some botnets, but HTTP is now more widespread, since it is even easier to implement and can be hidden in normal user navigation. There are other methods of communication that use covert channels (e.g., in DNS, ICMP etc.) [4]. The move from IRC-based architecture has happened because it was rather easy for ISP to disrupt botnet by blocking the central C&C. Considering the above weaknesses inherent to the centralized architecture of current C&C botnets, it is a natural strategy for botmasters to design a peer-to-peer (P2P) control mechanism into their botnets and in fact different kinds of P2P control architectures were implemented [21]. In a peer-to-peer architecture [12], there is no centralized point for C&C. Nodes in a peer-to-peer network act as both clients and servers such that there is no centralized coordination point that can be incapacitated. If nodes in the network are taken offline, the gaps in the network are closed and the network continues to operate

under the control of the botmaster. One more problem posed by P2P botnets to security specialists is the difficulty in estimating the size of the P2P botnet [22].

Botnets usually do not rely only on a single method of propagation but make use of a combined approach. Methods include scanning for vulnerable hosts [23], network shares, spam or unsolicited e-mail, P2P (Peer-to-peer), [4], net news, web blogs, other WEB resources [20], social engineering via an enticement 'lure' e-mail, browser exploit and malicious file download, [4], via instant messenger [22] and other common malware propagation methods. Separate botnet parts can use different propagation methods. In case botnet uses scanning for search of vulnerable hosts [23] three types of scanning can be separated: localized scanning (each bot chose the scanning range based on their own IP prefixes), targeted scanning (botmaster specifies a particular IP prefix for bots to scan) and uniform scanning (scanning the whole Internet).

Since information security community has accepted botnets as a threat and the use of countermeasures has been started the botnet creators were forced to use different obfuscation and deception methods to protect botnets and escape from punishment for illegal activities. The advanced modern botnet examples rely on a wide range of complex methods such as extremely resilient random topologies (including structured P2P networks), traffic anonymization [20], load balancing, reverse proxies for the C&C servers (making it harder to track down the attacks), fast-flux services (networks of compromised computer systems with public DNS records that are constantly changing), Rock Phish (compromised computers and thousands of DNS sub-domains are used in order to set up phishing scenarios that hide the real phishing site) [4], encrypted/obsfucated control channel [21] and many others [20], [21]. The recent trend is toward smaller botnets with only several hundred to several thousand zombies since big botnets are bad from the standpoint of survivability. It has also been suggested that the wider availability of broadband access makes smaller botnets as capable as the larger botnets earlier [24].

For better understanding of modern botnets further we provide a technical description of two botnet types: centralized, IRC based Agobot (description based on [25], [26], [27], [28]), other sources as cited and of the most famous P2P botnets STORM (description based on [29], [30]).

2.2 Botnet Examples

2.2.1 IRC Botnet – Agobot

Originally botnets were based on IRC and many of them still are. A botherder sets up an IRC server and clients connect to it. The server acts as the C&C from where the bots get their orders. All communication is based on the IRC protocol [31]. IRC is the most common botnet type because it is scalable and easy to hide within. While instances of botnets with looser control structures, such as those that use peer-to-peer networks, are increasing, IRC-style is still the most prevalent because it is scalable and provides instantaneous control over the bots [32]. In botnets that use the chat style of command and control, the attacker issues commands to the zombie hosts via a "rendezvous point," (Figure 1) which is usually an IRC server. The rendezvous point may or may not be a compromised machine since there are many public IRC servers that host unmonitored channels. The attacker and the zombie hosts subscribe to the same IRC channel. The attacker issues commands and the bots respond through that channel [3].

In June 1999 the first worm emerged to make use of IRC as a means of remote control. Written in Delphi, PrettyPark.Worm connected to a remote IRC server and allowed the attacker to retrieve a variety of information about the system. It also had a basic update mechanism which allowed it to download and execute a file from IRC.

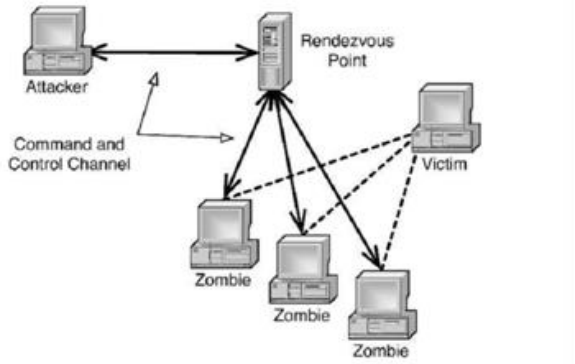

Figure 1: IRC botnet management scheme [3].

One of the most typical and widespread IRC botnets is the Agobot family (also known as Gaobot, Forbot, Phatbot, Urxbot, Rxbot, Rbot and compilations by individuals), which is among the most widespread in the number of variants created according to [25]. In some cases, there is an overlapping name, making the study of them very difficult [25]. The [25] paper presents a detailed overview of the Agobot family history and changes in functionality and description of them. Agobot is created on a modular basis and mainly affects computers, running MS Windows platform. Bot development kit is distributed under the GNU GPL license [25]. Agobot can exploit many well known OS vulnerabilities (e.g. buffer overflow) and back doors left by other viruses (a large collection of target exploits) [5]. Exploits and delivery functions are separated. Once the first step exploits succeed, it opens a shell on the remote host to download bot binary. The binary is encoded to avoid network-based signature detection. The bot has the module to test for debuggers (e.g. SoftIce) and VMWare once it is installed. If it detects VMWare it stops running. So VMWarebased Honeypot cannot run Agobot. [26]. The bots' functionality may include (depending on compilation) but is not limited to the following:

- credentials steeling, key-logging
- self-protection against Firewall/Antivirus processes by stopping them
- self-protection by blocking Antivirus Updates (by modification of HOSTS file)
- backdoor opening (using various listening ports), execution of commands and programs
- author notification about the Compromise via IRC
- commands acceptance via IRC
- IRC Client control interface protection by password
- remote update and deinstallation of the installed bot
- port scanning for detection of other vulnerable hosts
- DDoS functionality
- packet sniffing
- etc.

Botnets created on the Agobot framework mainly follow the centralized hierarchy, only some of them support the P2P communication channels.

2.2.2 P2P Botnet – STORM

As it was written earlier the STORM botnet which was linked to a single network with the help of the Storm Worm, special crafted Trojan horse, uses the decentralized hierarchy, as shown on Figure 2. We do not provide estimations on the number of infected hosts, since they differ a lot and are exaggerated in many cases, but [33] has named the Storm botnet as biggest security issue in 2008.

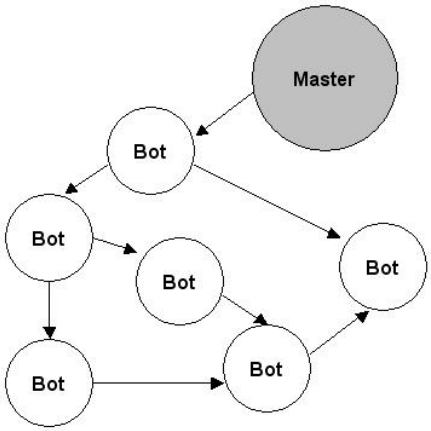

Figure 2: Decentralized P2P botnet.

There are no significant differences from the IRC botnets in malicious functionality, but it differs in resistance and self-protection mechanisms. The P2P-based botnet is very hard to trace and to shut down, because the botnet has robust network connectivity, uses encryption, and controls traffic dispersion. Each bot influences only a small part of the botnet, and upgrade/recovery is accomplished easily by its botmaster [29]. It has aggressive defences (it attacks anyone (DDoS) who tries to analyse it or reverse engineer it). It uses a clique architecture where each clique has its own 40-bit encryption key. There are no file exchanges between infected hosts which makes it difficult to track. StormWorm is now employing the same tactics as terrorist organisations. Each node belongs to a cell and knows only the members of the cell. If a cell is taken down, it is does not put the whole botnet at risk. When the botmaster needs to send information to all the nodes, he notifies one member of each cell who then passes on the information. This creates less traffic than if he were broadcasting to all the bots. Another advantage is that even if traffic is being monitored on a certain bot, suspicion is not raised if it keeps receiving messages from the same source [31].

Dissemination and functionality mechanisms are not so specific. Storm botnet first emerged in early 2007 and spread by sending e-mails with currently relevant subjects such as natural disasters and other topics in the news with attached or links to videos/images [31], [34]. It was spreading mainly using social engineering methods, like tricking people into downloading it from e-mails or websites. Storm was using fast-flux service networks. The website's DNS records changed every few minutes [4],[31]. It does not roam the Internet looking for vulnerabilities in machines that it can exploit [31].

After the executable has been downloaded and executed, it adds the system driver "wincom32.sys" to the Windows process "services.exe". Part of the installation involves hard-coding a peer list on the bot and saving it in a file called windir/system32/wincom32.ini. The Windows firewall is then disabled and several TCP ports are opened. The worm then bootstraps the bot onto the peer-to-peer Overnet network based on the Kademlia (eDonkey) algorithm so that it can contact its peer list if they are online [31]. Storm botnet, uses UDP-port 4000 for communication between peers. Such protocol makes closing down C&Cs – which would normally be an effective countermeasure against IRC botnets – useless [4]. After connecting to network and contacting its peer list, the bot is then ready for the secondary injections such as

Rootkit component, SMTP spamming component DDoS tool and etc. [31].
The success of Storm worm is partially due to the lack of security awareness in the average computer user. The other part of its success is the use of state-of-the-art technologies and a reputation for aggressiveness [31].

3.0 PRIOR AND RELATED WORK

Simulation environments serve many purposes, but they are only as good as their content. One of the most challenging and pressing areas that call for improved content is the simulation of bot armies (botnets) and their effects upon networks and computer systems [6].

3.1 General or Malware-Specific Models

Existing malware propagation models concentrate to forecasting the number of infected computers in the initial propagation phase (Figure 3).

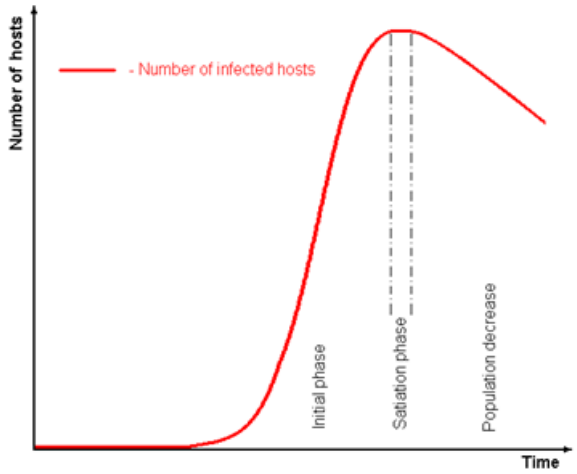

Figure 3: Internet worm propagation graph.

The first epidemiological model to computer virus propagation was proposed by [35]. Epidemiological models abstract from the individuals, and consider them units of a population. Each unit can only belong to a limited number of states. A SIR model assumes the Susceptible-Infected-Recovered state chain and SIS model – the Susceptible-Infected-Susceptible chain.

In a technical report [36] Zou et al. described a model of e-mail worm propagation. The authors model the Internet e-mail service as an undirected graph of relationship between people. In order to build a simulation of this graph, they assume that each node degree is distributed on a power-law probability function.

Malware propagation in Gnutella type Peer-to-Peer networks was described in [37] by Ramachandran et al. The study revealed that the existing bound on the spectral radius governing the possibility of an epidemic outbreak needs to be revised in the context of a P2P network. An analytical model that emulates the mechanics of a decentralized Gnutella type of peer network was formulated and the study of malware spread on such networks was performed.

The Random Constant Spread (RCS) model [38] was developed by Staniford et al. using empirical data

derived from the outbreak of the CodeRed worm. The model assumes that a machine cannot be compromised multiple times and operates several variables: *K* is the constant average compromise rate, which is dependant on worm processor speed, network bandwidth and location of the infected host; *a(t)* is the proportion of vulnerable machines which have been compromised at the instant t, *N•a(t)* is the number of infected hosts, each of which scans other vulnerable machines at a rate *K* per unit of time. But since a portion *a(t)* of the vulnerable machines is already infected, only *K•(1-a(t))* new infections will be generated by each infected host, per unit of time. The number *n* of machines that will be compromised in the interval of time *dt* (in which a is assumed to be constant). The model can predict the number of infected hosts at time *t* if *K* is known. The higher is *K*, the quicker the satiation phase will be achieved by worm. As [39] states, that although more complicated models can be derived, most network worms will follow this trend.

Other authors [40] propose the AAWP discrete time model, in the hope to better capture the discrete time behaviour of a worm. However, according to [41] continuous model is appropriate for large scale models, and the epidemiological literature is clear in this direction. The assumptions on which the AAWP model is based are not completely correct, but it is enough to note that the benefits of using a discrete time model seem to be very limited. On the other hand Zanero et al in [41] propose a sophisticated compartment based model, which treats Internet as the interconnection of autonomous systems, i.e. sub-networks. Interconnections are a so-called "bottlenecks". The model assumes, that inside a single autonomous system (or inside a densely connected region of an AS) the worm propagates unhindered, following the RCS model. The authors motivate the necessity of their model via the fact that the network limited worm Saphire which was using UDP protocol for propagation was following the RCS model till the "bottlenecks" were flooded by its scans.

Zou et al in [42] propose a two-factor propagation model, which is more precise in modeling the satiation phase taking into attention the human countermeasures and the decreased scan and infection rate due to the large amount of scan-traffic. The same authors have also published an article on modeling worm propagation under dynamic quarantine defence [43] and evaluated the effectiveness of several existing and perspective worm propagation strategies [44].

Ruitenbeek in [45] simulates virus propagation using parameterized stochastic models of a network of mobile phones, created with the help of Mobius tool and provides insight into the relative effectiveness of each response mechanism. Two models of the propagation of mobile phone viruses were designed to study the impact of viruses on the dependability and security of mobile phones: the first model quantifies the propagation of multimedia messaging system (MMS) viruses and the second - of Bluetooth viruses.

Lelarge in [46] introduces an economic approach to malware epidemic modeling (including botnets). Author states that users and computers on the network face epidemic risks. Epidemic risks (propagating viruses and worms in this case) are risks which depend on the behaviour of other entities (externalities) in the network. The model is based on graph theory and quantifies the impact of such externalities on the investment in security features in a network. Each agent (user) can decide whether or not to invest some amount to self-protect and deploy security solutions which decreases the probability of contagion. When an agent self-protects, it benefits not only to those who are protected but to the whole network. If all agents invest in self-protection, then the general security level of the network is very high since the probability of loss is zero. But a self-interested agent would not continue to pay for self-protection since it incurs a cost c for preventing only direct losses that have very low probabilities. When the general security level of the network is high, there is no incentive for investing in self-protection. This results in an under-protected network.

In [18] we have proposed the genetic algorithm based model, which was dedicated to evaluating existing as well as modeling other potentially dangerous Internet worms' propagation strategies at initial propagation phase. The efficiency of strategies was evaluated by applying the proposed fitness function.

The proposed model was tested on existing worms' propagation strategies with known infection probabilities. The tests have proved the effectiveness of the model in evaluating propagation rates and have shown the tendencies of worm evolution. We have also proposed the GA based propagation rate estimation model [47] which evaluated the negative (decrease) change of population size after satiation phase of a newly appearing worms by generating a decision tree based on a statistical data of known worms.

3.2 Botnet-Oriented Models

Several botnet-oriented models were proposed, but they all concentrate to tasks other than botnet evolution forecasting. As stated in [6] the development of botnet simulation and modeling capabilities requires advances in improving understanding of botnet technologies and the development of standards that support the simulation of bot army operations, but these tasks are complex for a variety of reasons, such as wide variety of botnets and their manner of propagation, challenge posed by modeling the amount of time and patterns of their infestation. That is why we say that Genetic algorithm approach for modeling botnets is extremely efficient due to its ability to solve complex problems with large solution space.

Sheila et al. in [6] use the epidemiological model as a basis for botnet modeling. The model is modified from this general model based upon the type of infection, transfer modality, and potential for re-infection and can be represented as a *M-S-E-I-R* chain, where *M* is the class of computers (hardware or software) who are not infected with malware that can be exploited to enable bot infestation; *S* is used to represent the class of computers that are infected during manufacture with malware that can be exploited to enable bot infestation. *E* is the set of computers that have been infected, are not transmitting the infection, and in whom the infection has not been detected; *I* is the set of computers that have been infected, are transmitting the infection, and in whom the infection has not been detected; *R* is the set of computers that have been infected, whose infection has been detected, and that have had their bot removed.

In their presentation Zou et al. [48] suggest using botnet propagation model via vulnerability exploitation and notice some similarities of bot and worm propagation. We can not agree with this statement since botnets use more propagation vectors than worms do. Botnet propagation modeling using time zones was proposed by Dagon et al. [49]. The model uses diurnal shaping functions to capture regional variations in online vulnerable populations. Authors of [50] have developed a stochastic model of P2P botnet formation to provide insight on possible defence tactics and examine how different factors impact the growth of the botnet.

Li et al. [51] model botnet-related cybercrimes as a result of profit-maximizing decision-making from the perspectives of both botnet masters and renters/attackers. From this economic model, they derive the effective rental size and the optimal botnet size. Fultz in [7] describes distributed attacks organized with the help of botnets as economic security games.

4.0 BOTNET STRATEGY EVOLUTION FORECASTING MODEL

When we try predicting the malware evolution trends, first of all we have to stand on the position of cybercriminals and evaluate their aims while developing malware. The turn point to malware commercialization was discussed previously in this article and currently there are no significant changes in this area. But shift in malware development tasks has also changed its development style. The developers now should evaluate the requirements of their "clients" such as robustness (how to generate a robust botnet even though some bots are removed), disclosure prevention (how to prevent significant exposure of the network topology even though some bots are detected), ease of use (how to easily monitor and obtain the complete information of a botnet by its botmaster, C&C user-friendly interface), protection of bots and botowners (how to prevent (or make it harder) defenders from detecting bots via their communication traffic patterns) [26], ensuring botnet growth and stability (propagation and survivability strategy

selection), functionality (botnets services) and others.
While creating the Genetic algorithm (GA) based model there are two main tasks a researcher has to solve: firstly, it is the correct selection of chromosomes and genes representing solution and secondly – creation of the fitness equation (or fitness evaluation criteria, such as statistical evaluation, etc.), which evaluates the fitness of the solution (single chromosome), generated during evolution and selects the most appropriate solutions according to the selection strategy. It is obvious, that solution representation is highly dependent on the analyzed problem and its complexity, and the fitness function – on the evaluated criteria.

The GA approach was shown to be effective while forecasting Internet worms propagation strategy evolution [18]. Although botnets are rather different malware type, GA use for their strategy forecasting seems promising since the algorithm itself simulates the natural evolution. In this chapter adoptions, extensions and modifications made to model first described in [18] are described. The botnet strategy evolution forecasting task is especially important to security specialists, working on countermeasures planning and implementation, since knowledge of botnet evolution trends allows selecting appropriate protection mechanisms.

4.1 Experiment Assumptions and Conditions

The main difference from the original model [18] is that due to lack of reliable statistical data botnet sizes, expansion speeds and effectiveness of different techniques used by botnets here we provide only the modeling framework, which includes botnet strategy representation chromosome description, which may be used to evaluate different modeling characteristics, such as survivability and defense, and the fitness function which may be used for propagation strategy evolution forecasting. No other changes to experiments conditions (population size, termination conditions, crossover conditions, mutation frequency, selection strategy) were implemented. This is mainly due to the fact that tuning these parameters may be effective only if using real statistical data. Here we provide only a proof of concept or a model that can be used later, when effective data collection methods about botnets will appear in scientific press.

GA consists of initialization, selection and evolution stages. During the initialization stage initial population of strategies is generated. Each strategy is represented as a chromosome. Initial population of strategies is generated on a random basis, i.e. each individual, representing separate botnet strategy is combined of random genes' values. At selection stage strategies are selected through a fitness-based process and in case termination condition is not met evolutionary mechanisms are started. In case termination condition is reached (>100 generations or evolution stagnation), algorithm execution is ended. If not – evolutionary mechanisms are activated. The crossover point for each pair of parents is selected randomly and defines the gene, after which the crossover operation is performed. The mutation operator defines the gene of a newly generated individual that should change value from current to any other random value from the range of possible gene values. Fitness proportionate selection is used for parent selection.

4.2 Strategy Representation

The behaviouristic characteristics of botnets (propagation vectors, communication channels and hierarchy, functionality, defense techniques) can be described as a botnet strategy. In GA modeling each strategy is represented as a chromosome. Botnets vary from other malware types in complexity. Many of them change behaviour at different periods of their existence. That is why creation of a universal representational chromosome is a complex task and a lot of changes should be made compared to the structure proposed in [18] for Internet worms. In fact we have to evaluate three different stages: propagation, satiation and resistance (the term we use to describe the period after satiation when the botnet faces the countermeasures or has to change the propagation strategy to overcome the satiation and start

growing again; during this phase botnet size stability, decrease or increase are possible, depending on the botnet selected strategy), i.e. for each stage a separate experiment is to be held. In order to insure inheritance between phases we propose using rather high rate of elitism (25% of the most effective strategies should be moved from one phase to another). This is a concept of elitism between phases, not generations.

The proposed strategy representation via chromosomes is provided in Table 1. Since compared to Internet worms botnets are more flexible (use parallel propagation methods, complex functionality, etc.) we propose using a reference system for gene activation and method definition, rather than static method description in a table, which would require to implement stricter limitations on number of methods (we allow selection from maximum 9 methods or gene deactivation). In Table 1 gene (fixed-length number – 10 positions) in a "Gene code" column just activates one or several methods from a reference database (samples are provided in the "Reference database (or sample)" column). "0" marks references to methods that do not exist, and other digits (1-9) references to the reference database. The number of non-zero digits shows the number of methods activated (used). There can not be the same non-zero digits in one gene (eg., 4510000700 is "OK", but 4550000700 is not). This check is performed during the initial population generation phase. Non-zero digit order is not important (this means that in case gene is 4510000700, then method 4, 5, 1 and 7 are active). In case of a 0000000000 gene this means that gene is not active and no methods are used. If gene is compulsory for botnet (e.g. for propagation) but is not active such an individual (strategy) will be simply eliminated by the evolutionary selection process). This check is performed during the initial population generation phase.

Table 1: Chromosome structure

NO./ GENE CODE GENE DESCRIPTION	**REFERENCE DATABASE (OR SAMPLE)**	**COMMENTS**
1 / TARGET_SEARCH Defines methods used by botnet for potential victim search	1) Scan – Random 2) Scan - Random, excluding 127.0.0.0/8, loopback, 224.0.0.0/8, multicast, LAN addresses 3) Scan - Random addresses from networks reserved for home user networks (DSL, etc.) 4) Non-automatic infection (i.e. manual) 5) E-mail spam – Malware / Link to malware 6) Instant messaging 7) Infected site 8) Removable media 9) P2P	Each method may use from 1 to 9 exploits. The necessary number of exploits with referencing exploiting probability are selected from the list of exploits. Limitations: exploits used for one method should run on a single platform.

2 / TRANSF Defines exploit body transfer mechanism	1) Connectionless (also called “Fire and forget”) 2) Connection oriented	Connectionless mechanism uses UDP protocol for exploit body transfer. Assumption is done, that it can fit in one datagram. Connection oriented mechanism uses TCP protocol for exploit transfer to the target host. For simplicity reasons, we make an assumption that both methods can be used, as for example attacking DNS (53/TCP,UDP) server. Without such an assumption additional check would be necessary. This is important in case botnet scans network for vulnerable hosts. In case other methods are used then they are always considered to be “Connection oriented”.
3 / EXPL_PLATF Defines the OS platform used for each method	1) DOS 2) *nix (Unix, Solaris, Linux) 3) Win9x 4) Win NT (NT, 2000, XP, Vista) 5) Mobile OS 6) Apple OS 7) Multi platform 8) Other OS 9) WEB application (mainly PHP) exploit	Despite the number of exploits used they all should run on a single platform. Different methods may run on different OS platforms. WEB based exploits are those that exploit vulnerabilities of WEB-based applications that do not depend on the OS. Mobile OS covers a variety of mobile OS (Symbian, WinMobile, iPhone OS, Android, Pal, etc.)
4 / EXPL_NUM Defines number of exploits used by each method	[0..9] (random number)	A random number of exploits used by each method, activated in TARGET_SEARCH. For example if TARGET_SEARCH gene is 4510000700 and EXPL_NUM is 1210000300, that means that the 4^{th} method is activated and it uses only one exploit, 5^{th} method uses 2 exploits, etc.

5 / HIERARCHY Defines the hierarchy used for communication	1) Central – one management host 2) Central – one management host – botnet is splited, when certain number of bots is achieved. After that operates as two (or more botnets) with different manament hosts, but managed by the same botmaster. The total number of bots in all botnets is calculated when evaluated. 3) Central – several (2-9) management hosts - independent 4) Central – several (2-9) management hosts – load balancing 5) Central – several (2-9) management hosts – fast-flux protection 6) Central – several (2-9) management hosts – fast-flux protection – load balancing 7) Belongs to several botnets (2-9) with central management hierarchy 8) Decentralized – P2P principal 9) Decentralized – P2P – fast-lux technology	In this gene only the first number out of 10 has sense and defines the hierarchy used for management. Other are just “0”.
6 / FUNCTIONALITY Defines the functionality that botnet provides for botnet owners.	1) Information collection (credentials steeling, keylogging) 2) Backdoor opening (using various listening ports), execution of commands and programs 3) Botnet owner notification about the Compromise; 4) Packet sniffing 5) DDoS functionality 6) Spam sending 7) Remote update and deinstallation of the installed bot 8) Botnet rental tools 9) Botnet management tools (ease-of-use)	This list is not complete since limited to the number of 9, which is selected for the reason of simplicity.
7 / SELF_PROTECT Defines methods that a single bot can use to protect itself against deinstalation, management blocking or analysis	1) Blocking Firewall/Antivirus processes 2) Blocking Antivirus Updates 3) Blocking OS updates 4) Deinstallation imitation, if detected by Antivirus with no real deinstallation 5) Imitation of usefulness (for eg., imitation of Antivirus – “Antivirus 2009”) 6) Period of inactivity 7) Low activity 8) De-installation if “honeypot” or “sand-box” or analysis is suspected 9) Social engineering (language selection, advertisements, related with latest or known events)	This list is not complete since limited to the number of 9, which is selected for the reason of simplicity.

In this article we do not analyze the case when some part of the botnet population (e.g. 35%) uses one strategy and another (65%) a bit or absolutely different.

It is also necessary to state that compared to Internet worms [18] there is a limited number of genes that can be "inactive". In *TRANSF_METHODS* at least method should be enabled, *TRANSF* – one method should be enabled, *EXPL_NUM* – at least one should be enabled, *EXPL_PLATF* – at least one should be enabled, *HIERARCHY* – one method should be enabled, *FUNCTIONALITY* – at least "Remote update and de-installation of the installed bot" should be enabled or any other method should be enabled, *SELF_PROTECT* – not compulsory if propagation or functionality evolution are being evaluated, but at least one is compulsory if botnet survivability is being evaluated.

4.3 Fitness Function

By definition the fitness function is a particular type of objective function that quantifies the optimality of a solution (i.e. an individual in a population) so that the particular individual may be ranked against all the other individuals. The fitnesis function depends on the botnets' behaviouristic characteristics we want to forecast. Here we propose fitness functions that allow forecasting botnet strategy evolution of population increase characteristics. It is obvious that the same chromosome structure and assumptions with different fitness function may be used for forecasting other characteristics, such as survivability, functionality, economic efficiency, etc. Our fitness function that is used for propagation strategy evaluation uses probabalistic and time comsumption parameters for methods activated by genes and 3 coefficents (non time-dependant).

From [38] we can say that the propagation strategy efficiency can be evaluated by value *K*, i.e. the number of computers the first malware individual in the wild can infect in a fixed time period. That means that the higher is *K*, the higher is the fitness of a propagation strategy. The fitness function we used is the following:

$$F(S) = k \cdot k_1 \cdot k_2 \cdot k_3 \cdot \left(1 - \prod_{i=1}^{9}\left(1 - p_i^1\right)\right) \cdot \left(1 - \prod_{i=1}^{9}\left(1 - p_i^2\right)\right) \cdot \left(1 - \prod_{i=1}^{9}\left(1 - p_i^3\right)\right) \cdot \left(1 - \prod_{i=1}^{9}\prod_{j=1}^{E[i]}\left(1 - p_{ij}^4\right)\right) \quad (1)$$

where:

F – the fitness function; *S* – strategy being evaluated; p_i^1 – probability, that the i^{th} method (*TARGET_SEARCH*) will find the live host; p_i^2 – probability, that the i^{th} method (*TRANSFER* method, corresponding to the i^{th} *TARGET_SEARCH* method) will successfully transfer the exploit to the target host; p_i^3 – probability, that the i^{th} method's (*EXPL_PLAT*) supported platform will be the same with the target host; p_{ij}^4 – probability, that the j^{th} exploit (out of *E[i]* exploits enabled by the i^{th} method (*EXPL_NUM*) will infect the target computer; *k* – the number of cycles the bot, using the evaluated strategy, can perform in one second time interval; k_1 – coefficient of the hierarchy effectiveness on the propagation rate; k_2 – coefficient of the functionality effectiveness on the propagation rate; k_3 – coefficient of the self protection functionality on the propagation rate. *k* is calculated according to the following expression:

$$k = \frac{1}{\sum_{i=1}^{7}\sum_{j=1}^{9} t_{ij}} \quad (2)$$

where t_{ij} are time expenditures needed for j^{th} method of the i^{th} gene, i=1..7, j=1...9.

Methods activated by genes 5-7 do not directly influence the propagation rate (except the case, that time is needed for their transfer and changes the *k* coefficient), since they do not carry any payload or provide any direct actions for propagation. But since we do not have enough statistical data regarding botnet propagation and influence of different functionality and organizational structure on the propagation rate

we cannot state, that these genes do not influence propagation at all. It is possible that some functions may increase or decrease the propagation rate due to some indirect qualities, for example hierarchy may be used to optimize the scan target, or blocking the AV would minimize the period of inactivity, etc. That is why we have introduced k_1, k_2, k_3 coefficients for *HIERARCHY*, *SELF_PROTECTION* and *FUNCTIONALITY*. These coefficients should be based on statistical evaluations and would be equal to 1, if no influence is detected, < 1 if influence is negative (slow the propagation rate) and > 1 if influence is positive.

So the fitness function can be read as: "The bot, using strategy S for propagation can perform *k* cycles per second, propagation rate is influenced by coefficients k_1, k_2 and k_3, which correspond to influence of hierarchy, functionality and self-protection features respectively. Propagation is successful if at least one of the target search methods finds the "live" target, the bot is successfully transferred to the target, at least one of the platforms, supported by the bot is found on the target, and at least one of all exploits at any supported platform resulted in infection."

4.4 Discussion and Future Work

The proposed model provides a general framework for evaluating different botnets' behaviouristic parameters (speed, survivability, manageability, etc.). Due to lack of the reliable statistical data we can only state that the proposed framework should be effective since rather similar model for Internet worm propagation strategy modeling was proved to be effective [18]. It is also necessary to notice that compared to the previous model [18] the model described here is more flexible and compact (possibility to use several target search methods; support of OS platforms in any combinations; increase of supported exploits to up to 81; the chromosome representation by digits is more universal and nearer to the "classical" GA representation; methods used by different genes are grouped in a compact manner), the model can be used to model botnet characteristics not only at different stages (propagation, satiation and resistance), but also with some limitations during the full lifetime period, i.e. not separated but interconnected phases due to introduction the elitism concept between phases. Although some limitations still exist (number of possible target search methods/exploits, assumptions that bot may use both TCP and UDP for transportation) other limitations, such as for probability rate limitations to 0.05 were removed, since the inverse probability ("at least one of the exploits" logic) was introduced.

The applicability of the framework is highly dependent on the collection of the reliable statistical data on botnet size and use of corresponding methods. Model tests on a real statistical data could be used for model tuning. Two more important modifications to the proposed model should be done in future to make it more flexible and realistic (although all models introduce some limitations and simplifications). The first is introducing the modeling possibility of populations, where exist several portions of population (for example 35% uses one strategy and another (65%) a bit or absolutely different) but the general fitness is evaluated against the whole population. The second is the evaluation of botnet strategy evolution tendency under pressure of countermeasures, as it was already done for Internet worms [52]. One more model extension opportunity is the creation of fitness functions for forecasting other botnet characteristics (e.g., survivability, manageability, etc.) or characteristic complexes (e.g. "propagation"+"survivability") evolution tendencies.

5.0 CONCLUSIONS

In this article the botnet-oriented extension to genetic algorithm based malware strategy evolution forecasting model was proposed, which aims at forecasting botnet propagation strategy evolution and may be used as a framework for other botnet characteristics evolution forecasting. The model consists of a propagation strategy representation structure, genetic algorithm acting under specified conditions and a fitness function for propagation strategy evaluation. Due to lack of the statistical data the model is described as a "proof-of-concept" with no real data tests.

Extensions made to the model allowed its applicability for a new malware type, i.e. botnets, made model framework more flexible and universal, several important limitations were eliminated. In order to use the model framework for different characteristic or characteristic complexes modeling only the fitness functions should be provided, leaving the chromosome structure untouched. The model framework may also be extended for modeling evolution under countermeasures and in other conditions, such as botnet populations combined of differently behaving parts.

The main model application area is countermeasures planning and scientific research of botnet evolution trends. The botnet strategy evolution forecasting task is especially important to security specialists, working on countermeasures planning and implementation, since knowledge of botnet evolution trends allows selecting appropriate protection mechanisms.

REFERENCES

[1] BRAND, Matthew; CHAMPION, Adam; CHAN, Derick (2009). *Combating the Botnet Scourge* [interactive]. [accessed 2009.11.26]. Link: <http://www.cse.ohio-state.edu/~champion/research/Combating_the_Botnet_Scourge.pdf>.

[2] PROVOS, Niels; HOLZ, Thorsten (2007). Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley Professional, 2007. 440 p. ISBN 0321336323.

[3] LEE, Wenke; WANG, Cliff; DAGON, David (2007). Botnet Detection : Countering the Largest Security Threat. Springer, 2007. 168 p. ISBN 0387687661.

[4] BARROSO, David (2007). Botnets - The Silent Threat. *ENISA Position Paper*, 2007, No. 3.

[5] BARFORD, Paul; YEGNESWARAN, Vinod (2005). An Inside Look at Botnets Proc. Special Workshop on Malware Detection. *Advances in Information Security*, 2005, p. 171-191.

[6] BANKS, B. Sheila; STYTZ, R. Martin (2008). Challenges Of Modeling BotNets For Military And Security. *SimTecT 2008 Refereed*, 2008.

[7] FULTZ, Neal (2008). Distributed attacks as security games. *Master thesis, US Berkley School of Information*, 2008.

[8] KARASARIDIS, Anestis; REXROAD Brian; HOEFLIN, David (2007). Wide-scale botnet detection and characterization. *USENIX Workshop on Hot Topics in Understanding Botnets*, 2007.

[9] LI, Zhichun; GOYAL, Anup; CHEN, Yan; PAXSON, Vern (2009). Automating Analysis of Large-Scale Botnet Probing Events. *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*. 2009, p. 11-12.

[10] RAJAB, Moheeb Abu; ZARFOSS, Jay; MONROSE, Fabian; TERZIS, Andreas (2007). My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 2007.

[11] JUKNIUS Jonas; ČENYS, Antanas (2009). Intelligent botnet attacks in modern Information warfare.*15th International Conference on Information and Software Technologies*, 2009, p. 39-42.

[12] GRIZZARD, B. Julian; SHARMA, Vikram; NUNNERY, Chris; KANG, Brent ByungHoon; DAGON, David (2007). Peer-to-peer botnets: overview and case study. *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 2007.

[13] RAJAB, Moheeb Abu; ZARFOSS, Jay; MONROSE. Fabian; TERZIS, Andreas (2006). A multifaceted approach to understanding the botnet phenomenon. *6th ACM SIGCOMM conference on Internet Measurment*, 2006, SESSION: Security and Privacy, p.41–52.

[14] WASH, Rick (2008). *Mental Models of Home Computer Security* [interactive]. [accessed 2009.03.12]. Link: <http://cups.cs.cmu.edu/soups/2008/posters/wash.pdf>.

[15] ZHUGE, Jianwe; HOLZ, Thorsten; HAN, Xinhui; GUO, Jinpeng; ZOU, Wei (2007) Characterizing the irc-based botnet phenomenon. *Peking University & University of Mannheim Technical Report*, 2007.

[16] SYMANTEC Corp. (2008). *Symantec Global Internet Security Threat Report Trends for July–December 07* [interactive]. [accessed 2009.03.12]. Link: <http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat report_xiii_04-2008.en-us.pdf>.

[17] STARNBERGER, Guenther; KRUEGEL, Christopher; KIRDA, Engin (2008). Overbot: a botnet protocol based on Kademlia. *SecureComm '08: Proceedings of the 4th international conference on Security and privacy in communication netowrks*, 2008, p. 1-9.

[18] GORANIN, Nikolaj, ČENYS, Antanas (2008). Genetic Algorithm Based Internet Worm Propagation Strategy Modeling. *Information Technology And Control*, 2008, Vol 37, No. 2, p. 133-140.

[19] HOLLAND, John (1975). Adoption in natural and artificial systems. The MIT press, 1975, 211 p.

[20] DAGON, David; GU, Guofei; ZOU, Cliff; GRIZZARD, Julian; DWIVEDI, Sanjeev; LEE, Wenke; LIPTON, Richard (2005). A Taxonomy of Botnets. *Proceedings of CAIDA DNS-OARC Workshop*, 2005.

[21] WANG, Ping; SPARKS, Sherri;.ZOU, C. Cliff (2007). An advanced hybrid peer-to-peer botnet. *USENIX Workshop on Hot Topics in Understanding Botnets*, 2007.

[22] DITTRICH, David; DITTRICH, Sven (2002). P2P as botnet command and control: A deeper insight. *In Proceedings of the 2008 3rd International Conference on Malicious and Unwanted Software*, 2008, p. 41-48.

[23] LI, Zhichun; GOYAL, Anup; CHEN, Yan (2007). Honeynet-based Botnet Scan Traffic Analysis. *Advances in Information Security* , 2007. p .25-44.

[24] VOGT, Ryan; AYCOCK, John; JACOBSON, Michael J.Jr (2007). Army of botnets. *Network and Distributed System Security Symposium*, 2007, p. 111-123.

[25] GORDON, John (2004). *Agobot and the „Kit" chen Sink* [interactive]. [accessed 2009.04.06]. Link: <http://www.infectionvectors.com/vectors/Agobot_&_the_Kit-chen_Sink.pdf>.

[26] ZHAOSHENG, Zhu; GUOHAN, Lu; YAN, Fu Chen; ZHI, Judy; Roberts, HAN, Phil, (2002). Botnet Research Survey. *COMPSAC '08: Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, 2008, p. 967-972.

[27] SYMANTEC Corp.(2004).*W32.Gaobot.DX | Symantec* [interactive]. [accessed 2009.05.20]. Link: <http://www.symantec.com/security_response/writeup.jsp?docid=2004-051816-5418-99>.

[28] STEWART, Joe (2004).*Phatbot Trojan Analysis* [interactive]. [accessed 2009.03.12]. Link: <http://www.secureworks.com/research/threats/phatbot>.

[29] ZHANG, Jun (2008). *Storm Worm & Botnet Analysis* [interactive]. [accessed 2009.04.11]. Link: <http://securitylabs.websense.com/content/Assets/Storm_Worm_Botnet_Analysis-June_2008.pdf>.

[30] KASPERSKY Lab (2008). *The botnet business* [interactive]. [accessed 2009.03.12]. Link: <http://www.viruslist.com/en/analysis?pubid=204792003>.

[31] MUKAMURENZI, Nelly Marylise (2008) Storm Worm: A P2P Botnet. *Master of Science in Communication Technology // Norwegian University of Science and Technology // Department of Telematics*, 2008.

[32] CANAVAN, John (2005). The Evolution of Malicious IRC Bots. *Proceedings of Virus Bulletin Conference 2005*, 2005, p. 104-114.

[33] SANS INSTITUTE (2008).*The sans institutes top ten cyber security menaces for 2008* [interactive]. [accessed 2009.05.20]. Link: <http://www.itu.int/osg/csd/newslog/The+SANS+Institutes+Top+Ten+Cyber+Security+Menaces+For+2008.aspx>.

[34] PORRAS, Phillip; SAIDI, Hassen; YEGNESWARAN, Vinod (2007). A Multi-perspective Analysis of the Storm (Peacomm) Worm. *Technical report, Computer Science Laboratory, SRI International*, 2007.

[35] KEPHART, O. Jeffrey; WHITE, R. Steve (1991). Directed-Graph Epidemiological Models of Computer Viruses. *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991, p. 342-359.

[36] ZOU, C. Cliff; TOWSLEY, Don; GONG, Weibo (2004). Email Virus Propagation Modeling and Analysis. *Technical report TRCSE- 03-04. – University of Massachussets,* 2004.

[37] RAMACHANDRAN, Krishna; SIKDAR, Biplab (2006). Modeling malware propagation in Gnutella type peer-to-peer networks. *Parallel and Distributed Processing Symposium,* 2006, Vol. 20, 8 pp.

[38] STANIFORD, Stuart; PAXSON, Vern; WEAVER, Nicholas (2002). How to 0wn the Internet in Your Spare Time. *Proceedings of the 11th USENIX Security Symposium,* 2002, p. 149-167.

[39] NAZARIO, Jose (2004). Defense and Detection Strategies against Internet Worms. Artech House, Inc, 2003. 322 p. ISBN 1580535373.

[40] CHEN, Zesheng; GAO, Lixin; KWIAT, Kevin (2003). Modeling the Spread of Active Worms. *Proceedings of IEEE INFOCOM 2003*, 2003, p. 1890-1900.

[41] SERAZZI, Giuseppe; ZANERO, Stefano (2004). Computer Virus Propagation Models. *Lecture Notes in Computer Science*, 2003, p. 26-50.

[42] ZOU, Cliff Changchun; GONG, Weibo; TOWSLEY, Don (2002). Code Red Worm Propagation Modeling and Analysis. *Proceedings of CCS'02*, 2002, p. 138-147.

[43] ZOU, Cliff Changchun; GONG, Weibo; TOWSLEY, Don (2003). Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense. *Proceedings of WORM'03*, 2003, p. 51-60.

[44] ZOU, Cliff Changchun; GONG, Weibo; TOWSLEY, Don (2005). On the performance of Internet worm scanning strategies. *Performance Evaluation*, 2006 Vol. 63, No. 7, p. 700-723.

[45] RUITENBEEK VAN, Elizabeth; COURTNEY, Tod; SANDERS, H. William; STEVENS, Fabrice (2007). Quantifying the Effectiveness of Mobile Phone Virus Response Mechanisms. *Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference*, 2007, p. 790-800.

[46] LELARGE, Marc (2002). The Economics of Malware: Epidemic Risks Model, Network Externalities and Incentives. *Fifth bi-annual Conference on The Economics of the Software and Internet Industries, Toulouse*, 2009.

[47] GORANIN, Nikolaj; ČENYS, Antanas (2008). Malware Propagation Modeling by the Means of Genetic Algorithms, Electronics and Electrical Engineering. *Electronics and Electrical Engineering*, 2008, No. 86. p. 23-26.

[48] ZOU, Cliff Changchun; DAGON, David; LEE, Wenke (2008). *Modeling and Measuring Botnets* [interactive]. [accessed 2009.05.01]. Link: <http://www.gtisc.gatech.edu/aroworkshop/ppt/BotnetModel_Zou.ppt>.

[49] DAGON, David; ZOU, Cliff, LEE, Wenke (2006). Modeling Botnet Propagation Using Time Zones. *In Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*, 2006.

[50] RUITENBEEK VAN, Elizabeth; SANDERS, William (2008). Modeling Peer-to-Peer Botnets. *Quantitative Evaluation of Systems, 2008. QEST '08. Fifth International Conference*, 2008, p. 307-316.

[51] LI, Zhen; LIAO, Qi Cliff; STRIEGEL, Aaron (2009). BotnetEconomics: Uncertainty Matters. Springer, 2009. p 1-23.

[52] GORANIN, Nikolaj; ČENYS, Antanas (2009). Genetic algorithm based Internet worm propagation strategy modeling under pressure of countermeasures. *Journal of Engineering Science and Technology Review*, 2009, No. 2, No. 88, p. 43-47.